AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) A method of generating an input file using a meta language regarding graphics data compression, the method comprising:

making an extensible markup language (XML) schema that defines at least a compression node describing object data to be compressed, and parameters used for data compression;

making style sheets which support conversion of an input XML file into a file, which is to be input to a data compression encoder, based on the XML schema; ~~and~~

generating the file, which is to be input to the data compression encoder, by parsing the input XML file based on the XML schema and the style sheets; and

inputting the generated file to the data compression encoder to generate an encoded bitstream of compressed graphics data.

2. (Previously Presented) The method of claim 1, wherein the XML schema further defines EncodingHints specifying the location of a file in which the object data to be compressed is stored.

3. (Original) The method of claim 1, wherein the parameters comprise at least one of a parameter for keyframe-based animation data regarding vertex coordinates of the object, a parameter for rotation information of the object, a

parameter for position information of the object to be compressed, and a parameter for three-dimensional (3D) mesh information to be compressed.

4. (Original) The method of claim 2, wherein the parameters comprise at least one of a parameter for keyframe-based animation data regarding vertex coordinates of the object, a parameter for rotation information of the object, a parameter for position information of the object to be compressed, and a parameter for three-dimensional (3D) mesh information to be compressed.

5. (Currently Amended) A method of generating an input file using a meta language regarding graphics data compression, the method comprising:

making an XMT schema which defines a compression node describing object data to be compressed, parameters for data compression, and BitWrapperEncodingHints which at least specifies a location of a file in which the object data to be compressed is stored;

making an XMT2BIFS style sheet which supports conversion of an XMT input file into a scene file and an XMT2MUX style sheet which supports conversion of the XMT input file into a mux file, based on the XMT schema; ~~and~~

generating the scene file and the mux file by parsing the input XMT file using the XMT schema and the XMT2BIFS and XMT2MUX style sheets, respectively; and

inputting the generated scene file and mux file to a data compression encoder to generate an encoded bitstream of compressed graphics data.

6. (Previously Presented) The method of claim 5, wherein the compression node comprises:

a node field comprising the object data to be compressed;

a buffer field which temporarily stores a bitstream defined in the compression node using an in-band scenario; and

a URL field which links information regarding the bitstream defined in the compression node using an out-band scenario,

wherein either the buffer field or the URL field is used.

7. (Previously Presented) The method of claim 6, wherein the compression node further comprises a type field specifying a type of node compression scheme.

8. (Original) The method of claim 5, wherein the parameters comprise at least one of a parameter for keyframe-based animation data regarding vertex coordinates of the object, a parameter for rotation information of the object, a parameter for position information of the object to be compressed, and a parameter for three-dimensional (3D) mesh information to be compressed.

9. (Previously Presented) The method of claim 5, wherein the BitWrapperEncodingHints further specifies an object descriptor ID that is the same as a URL ID of the compression node, a name of a file transmitting a compressed bitstream, and a type of a stream format, wherein the file name is described in the mux file.

10. (Previously Presented) The method of claim 5, wherein parsing the input XMT file further comprises:

receiving the input XMT file describing the compression node that defines the original data, compression parameters, and buffer; and

generating the scene file and the mux file by parsing the input XMT file using the XMT schema and the XMT2BIFS and XMT2MUX style sheets, respectively,

wherein the scene file comprises the object data to be compressed, the data compression parameters, and a buffer which temporarily stores a bitstream obtained from compression of the original data, and

the mux file describes a name of a file obtained by encoding the scene file using a BIFS encoder and a stream format.

11. (Previously Presented) The method of claim 5, wherein parsing the input XMT file further comprises:

receiving the input XMT file defining the compression node comprising a buffer temporarily storing the compressed object data; and

generating the scene file and the mux file by parsing the input XMT file using the XMT schema and the XMT2BIFS and XMT2MUX style sheets, respectively,

wherein the scene file comprises the buffer temporarily storing a bitstream that is a representation of the compressed object data, and

the mux file specifies a name of a file obtained by encoding the scene file using a BIFS encoder, and a stream format.

12. (Previously Presented) The method of claim 5, wherein parsing the input XMT file comprises:

receiving the input XMT file which describes the compression node defining the object data to be compressed, the data compression parameters, and URL information, and the BitWrapperEncodingHints defining an object descriptor ID that is the same as a URL ID of the compression node and a location of a bitstream that is a representation of the compressed object data; and

generating the scene file and the mux file by parsing the input XMT file using the XMT schema and the XMT2BIFS and XMT2MUX style sheets, respectively,

wherein the scene file comprises the object data to be compressed, the data compression parameters, and the URL information which links information regarding a bitstream obtained from compression of the object data, and

the mux file specifies the location of the bitstream, which is a representation of the compressed object data, and a stream format defined in the BitWrapperEncodingHints.

13. (Previously Presented) The method of claim 12, wherein the input XMT file further comprises an ObjectDescriptorUpdate which defines an object descriptor ID that is the same as the object descriptor ID specified in the BitWrapperEncodingHints, and a name of a the mux file to be generated from the parsing of the input XMT file,

wherein the scene file further specifies the object descriptor ID that is the same as the object descriptor ID specified in the BitWrapperEncodingHints, and the name of the mux file.

14. (Previously Presented) The method of claim 5, wherein parsing the input XMT file further comprises:

receiving the input XMT file describing the BitWrapperEncodingHints specifying the compression node defining a URL which links information regarding already-compressed object data, an object descriptor ID that is the same as a URL ID, and a location of a bitstream that is a representation of the compressed object data; and

generating the scene file and the mux file by parsing the input XMT file using the XMT schema and the XMT2BIFS and XMT2MUX style sheets, respectively,

wherein the scene file comprises the URL which links information regarding the bitstream obtained from the compression of the object data and whose ID is the same as the ID of the object descriptor specified in the compression node, and

the mux file specifies the location of the bitstream that is a representation of the compressed object data defined in the BitWrapperEncodingHints and a stream format.

15. (Previously Presented) The method of claim 14, wherein the input XMT file further comprises an ObjectDescriptorUpdate which specifies an object descriptor ID that is the same as the object descriptor ID defined in the BitWrapperEncodingHints and a name of the mux file obtained by the parsing of the input XMT file,

wherein the scene file further specifies the object descriptor ID that is the same as the object descriptor ID defined in the BitWrapperEncodingHints and the name of the mux file.

16. (Original) A computer readable recording medium for recording a program executing the method of claim 1 in a computer.

17. (Original) A computer readable recording medium for recording a program executing the method of claim 5 in a computer.

18. (Currently Amended) A system for generating an input file using a meta language regarding graphics data compression, the system comprising:

a processing unit which provides an XML schema that defines a compression node comprising at least information regarding object data to be compressed and parameters used for data compression;

a processing unit which provides style sheets which support conversion of an input XML file into a file which is to be input to a predetermined data compression encoder, based on the XML schema; and

an XML parser which parses the input XML file based on the XML schema and the style sheets to generate the file which is to be input to a predetermined data compression encoder.

19. (Original) The system of claim 18, wherein the parameters comprise at least one of a parameter for keyframe-based animation data regarding vertex coordinates of the object, a parameter for rotation information of the object, a parameter for position information of the object, and a parameter for three-dimensional (3D) mesh information to be compressed.

20. (Currently Amended) A system for generating an input file using a meta language regarding graphics data compression, the system comprising:

a processing unit which provides an XMT schema which defines a compression node specifying object data to be compressed, parameters for data compression, and BitWrapperEncodingHints which at least specifies a location of a file in which the object data to be compressed is stored;

a processing unit which provides an XMT2BIFS style sheet which supports conversion of an input XMT file into a scene file based on the XMT schema;

a processing unit which provides an XMT2MUX style sheet which supports conversion of the input XMT file into a mux file based on the XMT schema; and

an XMT parser which parses the input XMT file based on the XMT schema and the XMT2BIFS and XMT2MUX style sheets to generate the scene and mux files, respectively, as the input files to a predetermined compression encoder.

21. (Previously Presented) The system of claim 20, wherein the compression node comprises:

a node field specifying the object data to be compressed;

a buffer field which transmits a bitstream that is a representation of the compressed object data using an in-band scenario, the bitstream being specified in the compression node; and

a URL field which transmits the bitstream that is a representation of the compressed object data using an out-band scenario, the bitstream being specified in the compression node,

wherein the compression node uses either the buffer field or the URL field.

22. (Original) The system of claim 20, wherein the parameters comprise at least one of a parameter for keyframe-based animation data regarding vertex coordinates of the object, a parameter for rotation information of the object, a parameter for position information of the object, and a parameter for three-dimensional (3D) mesh information to be compressed.

23. (Previously Presented) The system of claim 20, wherein the BitWrapperEncodingHints further comprises an object descriptor ID that is the same as a URL ID defined in the compression node, a name of a file transmitting a compressed bitstream contained in the mux file, and a type of a stream format.

24-28. (Canceled)

29. (New) A method of generating an input file using meta representation of graphics data compression, the method comprising:

making an extensible MPEG-4 textual format (XMT) schema which defines a compression node describing object data to be compressed, parameters for data compression, and BitWrapperEncodingHints which at least specifies a location of a file in which the object data to be compressed is stored;

receiving an XMT input file;

generating a file to be input to a data compression encoder by parsing the input XMT file using information including the XMT schema; and

inputting the generated file to the data compression encoder to generate an encoded bitstream of compressed graphics data,

wherein the compression node comprises:

a node field containing the object data to be compressed;

a buffer field which is not used together with a URL field at the same time and temporarily stores a bitstream defined in the compression node using an in-band scenario;

the URL field which is not used together with the buffer field at the same time and links information regarding the bitstream defined in the compression node using an out-band scenario; and

a type field specifying a type of node compression scheme,

wherein the parameters comprise at least one of a parameter for keyframe-based animation data regarding vertex coordinates of the object, a parameter for rotation information of the object, a parameter for position information of the object to be compressed, and a parameter for three-dimensional (3D) mesh information to be compressed, and

wherein the BitWrapperEncodingHints further specifies an object descriptor ID that is the same as a URL ID of the compression node, a name of a file transmitting a compressed bitstream, and a type of a stream format, the file name being described in a mux file.

30. (New) A method of transmitting an input file generated using meta representation of graphics data compression, the method comprising:

making an extensible MPEG-4 textual format (XMT) schema which defines a compression node describing object data to be compressed, parameters for data

compression, and BitWrapperEncodingHints which at least specifies a location of a file in which the object data to be compressed is stored;

receiving an XMT input file;

generating a file to be input to a data compression encoder by parsing the input XMT file using information including the XMT schema; and

if the compression node of the XMT input file contains information about already-compressed object data and a buffer temporarily storing the already-compressed object data, transmitting a bitstream of compressed graphics data that is a representation of the already-compressed object data by using the buffer,

wherein the compression node comprises:

a node field containing the object data to be compressed;

a buffer field which is not used together with a URL field at the same time and temporarily stores a bitstream defined in the compression node using an in-band scenario;

the URL field which is not used together with the buffer field at the same time and links information regarding the bitstream defined in the compression node using an out-band scenario; and

a type field specifying a type of node compression scheme,

wherein the parameters comprise at least one of a parameter for keyframe-based animation data regarding vertex coordinates of the object, a parameter for rotation information of the object, a parameter for position information of the object to be compressed, and a parameter for three-dimensional (3D) mesh information to be compressed, and

wherein the BitWrapperEncodingHints further specifies an object descriptor ID that is the same as a URL ID of the compression node, a name of a file transmitting a compressed bitstream, and a type of a stream format, the file name being described in a mux file.

31. (New) A method of transmitting an input file generated using meta representation of graphics data compression, the method comprising:

making an extensible MPEG-4 textual format (XMT) schema which defines a compression node describing object data to be compressed, parameters for data compression, and BitWrapperEncodingHints which at least specifies a location of a file in which the object data to be compressed is stored;

receiving an XMT input file;

generating a file to be input to a data compression encoder by parsing the input XMT file using information including the XMT schema; and

if the compression node of the XMT input file contains information about already-compressed object data and a URL which links information regarding the already-compressed object data, transmitting a bitstream of compressed graphics data that is a representation of the already-compressed object data by using the URL,

wherein the compression node comprises:

a node field containing the object data to be compressed;

a buffer field which is not used together with a URL field at the same time and temporarily stores a bitstream defined in the compression node using an in-band scenario;

the URL field which is not used together with the buffer field at the same time and links information regarding the bitstream defined in the compression node using an out-band scenario; and

a type field specifying a type of node compression scheme,

wherein the parameters comprise at least one of a parameter for keyframe-based animation data regarding vertex coordinates of the object, a parameter for rotation information of the object, a parameter for position information of the object to be compressed, and a parameter for three-dimensional (3D) mesh information to be compressed, and

wherein the BitWrapperEncodingHints further specifies an object descriptor ID that is the same as a URL ID of the compression node, a name of a file transmitting a compressed bitstream, and a type of a stream format, the file name being described in a mux file.

32. (New) A method of transmitting an input file generated using meta representation of graphics data compression, the method comprising:

making an extensible MPEG-4 textual format (XMT) schema which defines a compression node describing object data to be compressed, parameters for data compression, and BitWrapperEncodingHints which at least specifies a location of a file in which the object data to be compressed is stored;

receiving an XMT input file;

generating a file to be input to a data compression encoder by parsing the input XMT file using information including the XMT schema; and

if the compression node of the XMT input file contains information about original data, compression parameters and a buffer, transmitting a bitstream of compressed graphics data by using the buffer, wherein the bitstream is obtained by compressing the original data using the compression parameters,

wherein the compression node comprises:

a node field containing the object data to be compressed;

a buffer field which is not used together with a URL field at the same time and temporarily stores a bitstream defined in the compression node using an in-band scenario;

the URL field which is not used together with the buffer field at the same time and links information regarding the bitstream defined in the compression node using an out-band scenario; and

a type field specifying a type of node compression scheme,

wherein the parameters comprise at least one of a parameter for keyframe-based animation data regarding vertex coordinates of the object, a parameter for rotation information of the object, a parameter for position information of the object to be compressed, and a parameter for three-dimensional (3D) mesh information to be compressed, and

wherein the BitWrapperEncodingHints further specifies an object descriptor ID that is the same as a URL ID of the compression node, a name of a file transmitting a compressed bitstream, and a type of a stream format, the file name being described in a mux file.

33. (New) A method of transmitting an input file generated using meta representation of graphics data compression, the method comprising:

making an extensible MPEG-4 textual format (XMT) schema which defines a compression node describing object data to be compressed, parameters for data compression, and BitWrapperEncodingHints which at least specifies a location of a file in which the object data to be compressed is stored;

receiving an XMT input file;

generating a file to be input to a data compression encoder by parsing the input XMT file using information including the XMT schema; and

if the compression node of the XMT input file contains information about original data, compression parameters, and a URL, transmitting a bitstream of compressed graphics data by using the URL, wherein the bitstream is obtained by compressing the original data using the compression parameters,

wherein the compression node comprises:

a node field containing the object data to be compressed;

a buffer field which is not used together with a URL field at the same time and temporarily stores a bitstream defined in the compression node using an in-band scenario;

the URL field which is not used together with the buffer field at the same time and links information regarding the bitstream defined in the compression node using an out-band scenario; and

a type field specifying a type of node compression scheme,

wherein the parameters comprise at least one of a parameter for keyframe-based animation data regarding vertex coordinates of the object, a parameter for

rotation information of the object, a parameter for position information of the object to be compressed, and a parameter for three-dimensional (3D) mesh information to be compressed, and

wherein the BitWrapperEncodingHints further specifies an object descriptor ID that is the same as a URL ID of the compression node, a name of a file transmitting a compressed bitstream, and a type of a stream format, the file name being described in a mux file.

34. (New) A method of generating an extensible MPEG-4 textual format (XMT) schema for use in meta representation of graphics data compression, the method comprising:

defining a compression node which includes information regarding object data to be compressed;

defining an encoding parameter required for data compression;

defining BitWrapperEncodingHints which at least specifies a location of a file in which the object data to be compressed is stored; and

storing the compression node, the encoding parameter, and the BitWrapperEncodingHints as the XMT schema,

wherein the compression node comprises:

a node field containing the object data to be compressed;

a buffer field which is not used together with a URL field at the same time and temporarily stores a bitstream defined in the compression node using an in-band scenario;

the URL field which is not used together with the buffer field at the same time and links information regarding the bitstream defined in the compression node using an out-band scenario; and

a type field specifying a type of node compression scheme,

wherein the parameters comprise at least one of a parameter for keyframe-based animation data regarding vertex coordinates of the object, a parameter for rotation information of the object, a parameter for position information of the object to be compressed, and a parameter for three-dimensional (3D) mesh information to be compressed, and

wherein the BitWrapperEncodingHints further specifies an object descriptor ID that is the same as a URL ID of the compression node, a name of a file transmitting a compressed bitstream, and a type of a stream format, the file name being described in a mux file.

35. (New) A system of generating an input file using meta representation of graphics data compression, the method comprising:

a processing unit which provides an extensible MPEG-4 textual format (XMT) schema for defining a compression node describing object data to be compressed, parameters for data compression, and BitWrapperEncodingHints which at least specifies a location of a file in which the object data to be compressed is stored; and

an XMT parser which generates a file to be input to a data compression encoder by parsing an input XMT file using information including the XMT schema,

wherein the compression node comprises:

a node field containing the object data to be compressed;

a buffer field which is not used together with a URL field at the same time and temporarily stores a bitstream defined in the compression node using an in-band scenario;

the URL field which is not used together with the buffer field at the same time and links information regarding the bitstream defined in the compression node using an out-band scenario; and

a type field specifying a type of node compression scheme,

wherein the parameters comprise at least one of a parameter for keyframe-based animation data regarding vertex coordinates of the object, a parameter for rotation information of the object, a parameter for position information of the object to be compressed, and a parameter for three-dimensional (3D) mesh information to be compressed, and

wherein the BitWrapperEncodingHints further specifies an object descriptor ID that is the same as a URL ID of the compression node, a name of a file transmitting a compressed bitstream, and a type of a stream format, the file name being described in a mux file.

36. (New) The system of claim 30, wherein, if the compression node of the XMT input file contains information about already-compressed object data and a buffer temporarily storing the already-compressed object data, the encoder transmits a bitstream that is a representation of the already-compressed object data by using the buffer.

37. (New) The system of claim 30, wherein, if the compression node of the XMT input file contains information about already-compressed object data and a URL which links information regarding the already-compressed object data, the encoder transmits a bitstream that is a representation of the already-compressed object data by using the URL.

38. (New) The system of claim 30, wherein, if the compression node of the XMT input file contains information about original data, compression parameters, and a buffer, the encoder transmits a bitstream by using the buffer, wherein the bitstream is obtained by compressing the original data using the compression parameters.

39. (New) The system of claim 30, wherein, if the compression node of the XMT input file contains information about original data, compression parameters, and a URL, the encoder transmits a bitstream by using the URL, wherein the bitstream is obtained by compressing the original data using the compression parameters.